Transformations

Geometry vs. Topology

Geometry: Where things are (e.g., coordinates)
Topology: How things are connected

Geometry = changed
Topology = same (1-2-3-4-1)

Geometry = same
Topology = changed (1-2-4-3-1)

Transformations

Suppose you have a point P and you want to move it over by 2 units in X. How would you change P's coordinates?

\[ P' = (P_x + 2, P_y, P_z) \]

This is known as a coordinate transformation

General Form of 3D Linear Transformations

\[ x' = Ax + By + Cz + D \]
\[ y' = Ex + Fy + Gz + H \]
\[ z' = Ix + Jy + Kz + L \]

It's called a “Linear Transformation” because all of the coordinates are raised to the 1st power, that is, there are no \( x^2 \), \( x^3 \), etc. terms.

Transform the geometry – leave the topology as is
Scaling About the Origin

\[ x' = x \cdot S_x \]
\[ y' = y \cdot S_y \]
\[ z' = z \cdot S_z \]

2D Rotation About the Origin

\[ x' = x \cos \theta - y \sin \theta \]
\[ y' = x \sin \theta + y \cos \theta \]

Linear Equations in Matrix Form

\[ x' = Ax + By + Cz + D \]
\[ y' = Ex + Fy + Gz + H \]
\[ z' =Ix + Jy + Kz + L \]

Identity Matrix (\([ I ]\))

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

Matrix Inverse

\[ [M] \cdot [M]^{-1} = [I] \]
\[ [M] \cdot [M]^{-1} = "Nothing has changed" \]

Translation Matrix

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]
### Scaling Matrix

\[
\begin{bmatrix}
S_x & 0 & 0 & 0 \\
0 & S_y & 0 & 0 \\
0 & 0 & S_z & 0 \\
1 & 0 & 0 & 1
\end{bmatrix}
\]

Quick! What is the inverse of this matrix?

### 3D Rotation Matrix About Z

\[
\begin{pmatrix}
\cos \theta & -\sin \theta & 0 & 0 \\
\sin \theta & \cos \theta & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

Right-handed coordinates

Right-handed positive rotation rule

+90º rotation gives: \( y' = x \)

### 3D Rotation Matrix About Y

\[
\begin{pmatrix}
\cos \theta & 0 & \sin \theta & 0 \\
0 & 1 & 0 & 0 \\
-\sin \theta & 0 & \cos \theta & 0 \\
1 & 0 & 0 & 1
\end{pmatrix}
\]

+90º rotation gives: \( x' = z \)

### 3D Rotation Matrix About X

\[
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & \cos \theta & -\sin \theta & 0 \\
0 & \sin \theta & \cos \theta & 0 \\
1 & 0 & 0 & 1
\end{pmatrix}
\]

+90º rotation gives: \( x + y \)

### Compound Transformations

Q: Our rotation matrices only work around the origin. What if we want to rotate about an arbitrary point (A,B)?

A: Use more than one matrix.

Write it

Sey it

http://xkcd.com
Matrix Multiplication is not Commutative

Matrix Multiplication is Associative

Can Multiply All Geometry by One Matrix!

OpenGL Will Do the Transformation Compounding for You!

The Funky Rotation Matrix for an Arbitrary Axis and Angle
The Funky Rotation Matrix for an Arbitrary Axis and Angle

\[ P' = P \]
\[ P' = P \cos \theta + Q \sin \theta \]
\[ P' = P \times P' \]

\[ P' = [\hat{A} \dot{P}] + \cos \theta [P - \hat{A}(\hat{A} \dot{P})] + \sin \theta [\hat{A} \times P] \]

For this to be correct, A must be a unit vector.